DRAFT TWG-98-45

Federal PKI Client "Wish List"

July 31, 1998

CYGNACOM SOLUTIONS

1 Introduction

The Federal PKI will not be a monolithic, structure, nor will it be a single enterprise PKI. Rather it will be knit together from numerous agency PKIs, some initially limited to particular applications, and some agency-wide (or enterprise scale) multi-application PKIs. These will use CA products or services from different vendors, as well as client products from many vendors. The approach adopted for the Federal PKI is based on the concept of a “bridge CA,” that provides trust (or certification) paths between “principal CAs” in each agency. Industry organizations and other nations are adopting similar solutions, where a designated CA cross-certifies with high level CAs in different trust domains, to create certification paths. This approach will allow a large-scale government, industry, national or global PKI to be assembled from application or enterprise scale PKIs. The architecture is illustrated in Figure 1. The approach is further described in [TWG-98-29] and the Federal PKI Concept of Operations [CONOPS].

However, for the bridge CA approach to work, clients need greater capabilities than are required for a simple application oriented PKI, or even an enterprise CA. It does little good to create a PKI with extensive certification paths unless client applications can build and process the certification paths. While some currently available products do offer extensive capabilities to find and process certificate paths, as defined in the X.509 standard, many, perhaps most do not. This paper outlines the client features that the Federal PKI Technical Working Group feels are needed to use the bridge CA concept, in order to have a large scale PKI.

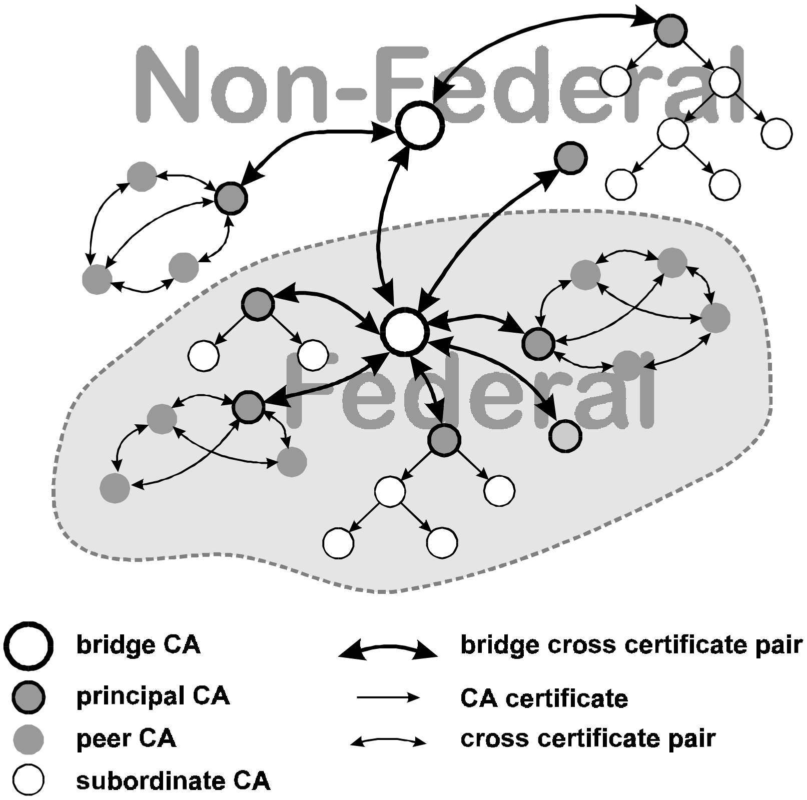

Figure 1 - The FPKI Certification Path Architecture

1.1 Scope

The following features list only applies to client-based products. Client-based products include (but are not limited to) products such as Certification Authority Workstations, Web Servers and Browers, and Electronic Messaging applications.

1.2 Infrastructure Features

In the near future, two additional wish lists will be developed. These two wish lists will cover Certificate Authority and directory features required providing an infrastructure, which supports client applications in path development, and processing.

One Wish List will cover cross-enterprise interoperability requirements between Certificate Authorities. The other Wish List will detail the features required of repositories (X.500, LDAP).

1.3 Approach

The document is divided into two sections: Primary Features and Secondary Features. The Primary Features section lists all features the FPKI-TWG would like integrated immediately in vendor products. The Secondary Features list describes ancillary features that the TWG would like to see implemented in vendor products.

2 Primary Features

The FPKI would like to the following features implemented by vendors as soon as possible, because secure cross-vendor, cross enterprise public key interoperation cannot occur without them:

1. Certificate Policy Management
2. Developing Certificate Paths
3. Validating Certificate Paths
4. Support Multiple Signature Algorithms
5. Certificate Revocation List Support
6. Lightweight Directory Access Protocol

2.1 Certificate Policy Management

Different applications call for different security practices. Users need to be able to specify acceptable policy requirements for any given application.

This feature would give Users the capability to define the acceptable certificate policies for an application. I.e. – a User can specify that the “DoD Medium policy” and “Federal High” certificate policies are the only acceptable policies for a given application. Many Federal public key users rely on certificate policies to determine which certificates may be used for specific applications. In the absence of support for policy management features, the only way to control which policies are accepted is to maintain separate, non-interconnected networks, or link different certificate policies to separate CAs, then maintain a list of acceptable CA’s. Neither of these approaches provides the required level of security and interoperability.

2.2 Developing Certificate Paths

The initial step to verifying a digital signature is to build a trusted path from a subscriber certificate to the relying parties trusted CA.

Commercial products must have the ability to build certificate chains. Client application need to build certificate paths to validate signatures and certificates as defined in the Federal PKI X.509 Certificate and CRL Extensions Profile [X509_PROFILE] (Section 1.3 – Certificate Path Development). This development procedure requires access to certificates. I.e.- Retrieving certificates from a cache or getting certificates from a repository.

Most of the client applications available today do not have the capability to build cross-enterprise certificate chains. This mainly applies to current Web Browers and any client applications that do not support certificate path development. Large, cross-enterprise infrastructures generally require certificate chains. Applications that cannot develop and process certificate chains including multiple intermediate certificates are not suitable for use in such large infrastructure applications.

2.3 Validating Certificate Paths

The second portion of validating a signature involves processing the certificate chain from the subscribers certificate to the relying parties trusted CA.

Client applications need to process certificate paths to validate signatures and certificates as defined in the X.509 Standard (Section 12.4.3 – Certification path processing procedure) and Federal PKI X.509 Certificate and CRL Extensions Profile (Section 1.4 Certification Path Processing Procedure). This mainly applies to current Web Browers and any client applications that do not support full path processing.

The security of public key crypto-systems can be no more robust than the processing of the certificates. Applications must correctly process all certificate components, including all standard certificate and Certificate Revocation List (CRL) extensions in order for cross-enterprise security services to be provided. For example, certificates containing extensions that the ITU X.509 standard requires or allow to be critical cannot be processed by applications not implementing those extensions – but some of these extensions provide little benefit unless marked critical.

The Federal PKI X.509 Extensions Profile [X509_PROFILE] requires that the following certificate extensions must be processed:

- keyUsage
- certificatePolicies
- policyMappings
- subjectAltName
- issuerAltName
- basicConstraints
- nameConstraints
- policyConstraints
- cRLDistributionPoints

The certificate extensions are further described in [X509_PROFILE].

2.4 Support Multiple Signature Algorithms

This feature allows users to verify signatures with any of the widely accepted signature algorithms. This allows users to verify signatures applied by a broader range of entities and applications.

Currently, the only signature algorithm that most vendors support is RSA or DSA. It is uncommon to find that both are supported. Such algorithms should include RSA, Digital Signature Algorithm (DSA), and Elliptic Curve DSA (ECDSA).

At a minimum, RSA, DSA, and ECDSA must be supported. Also, the signature algorithm OID’s listed by the IETF must be supported by client applications.

2.5 Certificate Revocation List Support

Users and applications need to know whether a given certificate is still valid.

CRL’s should be supported as defined in the FPKI X.509 Certificate and CRL Extensions profile [X509_PROFILE]. At a minimum, client applications must be able to retrieve CRL’s, process CRL’s, and determine whether or not they are full CRL’s.

The Federal PKI X.509 Extensions Profile [X509_PROFILE] requires that the following CRL extensions be processed:

- issuerAltName
- issuingDistributionPoint

The CRL extensions listed above are further described in [X509_PROFILE].

2.6 Lightweight Directory Access Protocol (LDAP)

Users and applications must be able to locate certificates and CRL's for certificate path development and validation.

Vendor products must support retrieving certificates and/or CRL's via requests to X.500 directories. The protocol used to query the directories must be LDAP (with referrals).

3 Secondary Features

These features are not critical to interoperability between the Federal world and the rest of the community. However, the FPKI wishes to see these features implemented in vendors' products, as they will be very useful in the future.

The following is a list of features the FPKI-TWG would like to see implemented into vendor products after they have implemented the requested "Primary" features:

1. Key Management
2. Support Multiple Symmetric Key Algorithms
3. FIPS 140-1 Cryptographic Module Validation

3.1 Key Management

Vendor applications must have common key exchange/agreement algorithms.

This feature is required to provide client applications with the capability to use asymmetric key management algorithms for exchanging common symmetric keys. At a minimum, the key management algorithms that must be supported include ANSIX9.42 based Diffie-Hellman (key agreement), RSA (key transfer), and Elliptic Curve Diffie-Hellman (ECDH) (key agreement).

3.2 Support Symmetric Key Algorithms

Vendor applications need to support multiple symmetric key algorithms.

Vendors support of multiple symmetric key algorithms will enable users to achieve confidentiality of communication across infrastructures. Symmetric key algorithms should only include FIPS approved algorithms. Currently, only DES and Skipjack are FIPS standards.

3.3 FIPS 140-1 Cryptographic Module Validation

Many medium and high assurance applications require use of FIPS 140-1 approved cryptographic modules – particularly for certificate signing.

This would require that vendor's have all cryptographic modules FIPS 140-1 approved. Compliance testing also includes DES (FIPS 46-2), SkipJack, DSS (186), and SHA-1 (FIPS 180-1) algorithms. Perhaps in the future, other algorithms will be included as well; RSA, ECDSA, AES, triple-DES, etc.

References

[CONOPS] TWG-98-31, *Draft Federal PKI Concept of Operations*, 3 June 1998

[TWG-98-29] W. E. Burr, “Proposed Federal PKI Architecture,” 19 May 1998

[X509_PROFILE] TWG-98-07, Federal Public Key Infrastructure (PKI) X.509 Certificate and CRL Extensions Profile